UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques
Universitat de Barcelona

Data Science for a New Generation of Tutors: Building an Academic-Guidance System Based on Dropout and Grades Prediction

Autor: Sergi Rovira Cisterna

Director:	**Dra. Laura Igual**
Realitzat a:	**Departament de Matemàtiques i Informàtica**

Barcelona, **26 de gener de 2017**

Contents

Acronyms

EAG Education at a Glance

EHEA European Higher Education Area

SVM Support Vector Machines

LR Logistic Regression

RF Random Forest

AdaBoost Adaptive Boosting

GB Gaussian Naive Bayes

SVR Support Vector Regression

SMOTE Synthetic Minority Over-sampling Technique

bagging Bootstrap aggregating

PCA Principal Component Analysis

MAE Mean Absolute Error

Abstract

This work is part of an innovative educational project which aim is to create a tool to help tutors offer more personalised and proactive guidance to the students. An analysis of the performance of different Machine Learning techniques for dropout intention prediction is presented. The approach of using Recommender Systems for final grade prediction and course ranking creation has been also assessed. Visualizations which help in the interpretation of the obtained results have been developed and a design for the tutoring tool has been outlined. The research has been performed using data from the degree studies in Law, Computer Science and Mathematics of Universitat de Barcelona.

Resum

Aquest treball és part d'un sistema d'innovació docent que té com a objectiu crear una eina per ajudar al tutor a oferir ajuda als estudiants de forma més proactiva i personalitzada. S'ha analitzat l'ús de diferents tècniques de Machine Learning per a la predicció d'intenció d'abandonament. També s'ha avaluat l'ús de sistemes de recomanació per a la predicció de notes i creació de rànquing d'assignatures. S'han desenvolupat visualitzacions que permeten interpretar els resultats com també s'ha proporcionat un possible disseny de l'eina. L'estudi s'ha realitzat utilitzant dades d'estudiants dels graus de Dret, Enginyeria Informàtica i Matemàtiques de l'Universitat de Barcelona.

Resumen

El presente trabajo es parte de un sistema de innovación docente que tiene como objetivo crear una herramienta para ayudar al tutor para que pueda ofrecer ayuda a los estudiantes de forma mas proactiva y personalizada. Se ha analizado el uso de diferentes técnicas de Machine Learning para la predicción de intención de abandono. También se ha evaluado el uso de sistemas de recomendación para la predicción de notas y creación de ránking de asignaturas. Se han desarrolado visualizaciones que permiten interpretar los resultados y se ha proporcionado un posible diseño de la herramienta. El estudio se ha realizado utilizando datos de estudiantes de los grados de Derecho, Ingeniería Informática y Matemáticas de la Universitat de Barcelona.

Chapter 1

Introduction

Since Bologna Process [8] was introduced, most European Universities have developed a tutoring system to provide their students with mentorship. The responsibilities of the tutor may differ between institutions but his/her main role is to offer personal guidance and advice to the students. Several recent works point out that personalized mentorship is crucial to prevent dropout, and improve their academic performance. Actually, decreasing dropout is one of the main current goals in European Higher Education Area (EHEA). The European dropout rate is 30% according to the publication Education at a Glance (EAG) [1]. In Spain, the dropout rate stands between 25% and 29% according to [25] and in the University of Barcelona, the dropout rate from 2009 to 2014 was around 20% accordingly to [2].

Moreover, universities now offer a broader range of specialized degrees than ever before (minors, double degrees, interdisciplinary and inter-university masters). Therefore the number and variety of students has increased and consequently has made tutorship a more challenging task. All data recorded for every student, such as grades, hours of study and previous academic achievements can be useful information for the tutor but it is not always available. Even if this data is gathered and the tutors have access to it, the sheer size of information is unmanageable by them.

In this context, an automatic tool to process and analyze the accumulated annual curricular data of the students could be extremely helpful for the tutor task [17, 30]. In this work, we present a data-driven system, based on machine learning techniques, for two different tasks: 1) the early prediction of student dropout and 2) the prediction of subsequent course grades for every student, as well as personalized course recommendations. The early dropout prediction indicates those students who are in most need of help. Tutors can focus on them and thus, increase their motivation and performance. Moreover, the course final grade predictions and course recommendations are all information useful to provide personalized enrollment guidance and orientation. Tutor can provide information on the courses that a particular student may enroll on each academic year which will most likely result in success.

For the first task, we compare five state-of-the-art classifier methods: Logistic Regression (LR) [22], Gaussian Naive Bayes (GB) [24], Support Vector Machines (SVM) [7], Random Forest (RF) [16] and Adaptive Boosting (AdaBoost) [15], with the aim of provid-

ing as much inside to the techniques as possible. For the second task, we compare three methods: Collaborative Filtering Recommendation System [20], Linear Regression [31] and Support Vector Regression (SVR) [27]. We extensively validate the proposed methods and select the approach with the best performance. We obtain promising results for the degree studies in Law, Computer Science and Mathematics at the University of Barcelona. We also present new visualizations for the interpretation of the different results, which includes student trends in behavior and academic preferences, providing a rich seam of information to tutors and heads of departments in universities. Also, this would enable them to take immediate action to improve their students' welfare and academic performance which in turn, would prevent students dropping out.

The presented system, techniques and visualizations in a tutor tool for evaluating dropout intention and predicting grades which can be easily adapted to any degree study and updated annually. This tool unveils information about the students, therefore it must be confidential and restricted to tutors and heads of department. This limitation tries to avoid any stigmatization of the students by their professors.

To the best of our knowledge, this is the first work which applies several machine learning techniques to predict academic grades and dropout. Previous work has focused on statistical approaches to study the dropout (see Section 1.1). Statistical models are based on assumptions drawn from the underlying problem. If these assumptions are wrong the predictive power of the model will be poor. A statistical analysis is superior to machine learning techniques when trying to understand the variables involving the problem. However, machine learning models are better when it comes to predictive performance because they are not based on assumptions over the problem but over the provided data. Adaptability is another advantage of machine learning techniques over statistics. Taking the dropout problem into consideration, if student performance factors vary over time (difficulty of the courses for example) the assumptions of a statistical model could become obsolete. However, a machine learning model would easily adapt to the new data.

1.1 Related Work in Education Science

Several recent works study the causes related to dropout intention [3–5, 9, 10, 12, 13, 21, 25, 29]. In the majority of these works the dropout rate is defined as the number of students who register for a course and did not formally enroll again for the next two consecutive academic years. This definition is also used in the University of Barcelona and in this work. Paper [3] states that the study of dropout should take into account two different situations: leave the university system or withdraw from the actual studies but changing to another faculty or institution. More precisely, this distinction does not allow us to better identifying students that have problems with their studies than using the first definition.

The aforesaid studies use data from different sources: public databases such as UNEIX (Portal of the Information System for Universities and Research in Catalonia) [13, 25], data from a particular university [4, 12], or collected data by means of ad-hoc interviews/questionnaires [4,5,9,10,12,13,25]. These data contains different information: from student demographic characteristics and educational resources to personal opinions on different academic regards. Several of these studies analyzed data from Catalan universities [9,10,13,25].

Collecting personal data, other than academic performance, can be useful for predicting dropout intention, but it may be also costly to gather. In our approach, we train our models using the final grade of each course because this data is already tracked by the universities. Moreover, this information is updated periodically and can be taken into account by our models. However, any other kind of valuable information can be easily added to the models in case of it being provided by the university.

In most of previous studies the variables considered to be predictive of dropout intention are related to the student educational background, his/her actual performance at the university and socioeconomic factors. A wide range of approaches are used to identify and validate the importance of such variables for dropout intention prediction. The author of [29] was the first to focus on the dropout problem and encourage the research on this issue. In [4], a statistical descriptive analysis is used. This study concludes that previous academic performance, first year academic performance, class attendance and enrollment date are variables that are directly linked to dropout. In [9, 10] the authors study dropout intention and learning outcomes simultaneously and create a conceptual model that directly relate the two concepts. The conclusion drawn from this research is that the level of academic satisfaction is important to predict dropout intention. Another approach is adopted in [14] where the authors perform dropout intention prediction using logistic regression with categorical variables such as level of studies of the parents, parents occupation, sex and first year academic performance. Our contribution uses also machine learning techniques, but comparing five different classification models to predict dropout intention.

After analyzing the explanatory indicators of dropout intention, the studies mentioned above suggest different actions that could be performed to reduce dropout rates. For instance, according to the authors of [4, 12], fomenting class attendance and participation, collecting and storing information of the students and developing a program for new students are essential tasks that a university could perform to reduce dropout rates. In [9,10]

it is mentioned that increasing the level of satisfaction with the university experience and the cognitive outcomes would help to reduce dropout rates. To focus on the quality of educational resources and lectures as well as seeking a realistic expectation held by the students before matriculation are among the main tasks to reduce dropout rates. The study in [13] states that there exists moments of special relevance when facing the decision to drop out and that there is a need to provide personal and academic guidance to the students. In [25], the study concludes that an improvement of vocational counseling practices along with mentorship programs would benefit universities and reduce dropout rates. Similar suggestions are stated in other studies such us [5]. The tool presented in this work could assist universities to implement these suggestions more easily.

Our study is also related to previous works on the educational field as the one presented in [28], where several prediction techniques in data mining are implemented to assist educational institutions with predicting students' grade averages at graduation time; and the study in [11], which identifies some of the factors that influence the probability of successfully pass a first course in Mathematics by using a classic logistic regression model (logit) and, an asymmetric Bayesian logit model. Two other related works are those presented in [17] and [30]. The former offers a data-driven system to personalize the communication between students and instructors for large STEM (*Science, Thechnology, Engeneering and Mathematics*) introductory courses. The latter studies the difference in motivation and academic self-concept between first-year college students of STEM courses depending on their gender.

1.2 Context of the project

The present work is part of the teaching innovation project created in *Departament de Matemàtiques i Informàtica* and *Departament de Mètodes de Investigació i Diagnòstic en Educació (MIDE)*.

The final aim of this project is to create a tool to help the tutor.

The project has been divided into five phases:

1. Acquisition, centralization and anonymization of student's data.
2. Data exploration using data science and statistical techniques.
3. Use of machine learning techniques to predict final students grades.
4. Development of the system.
5. Evaluation of the system.

   This project is part of phases 2, 3 and 4.

   We have been working in this project since January 2016 with the financial support of the University of Barcelona (Grant 2014PID-UB/068). The project has led to publication [26].

Chapter 2

Methodology

2.1 Data Gathering

To conduct our research, we have gathered data from a total of 4,434 students who studied the degree in Law (3,463), Mathematics (516) or Computer Science (455) in the University of Barcelona (UB) between the years 2009 and 2014.

The gathered data consists of the variables in the table 2.1.

Notes	Any primera matricula	Universitat procedencia
Becat	Nota acces	Sistema educatiu estranger
Sexe	Id via acces	Pais sistema estranger
Naixement	Lloc secundaria	Lloc cfgs
Nacionalitat	Tipus lloc secundaria	Tipus lloc cfgs
Simultaneitat	Priv pub secundaria	Priv pub cfgs

Table 2.1: **Variables of the datasets**

Both the degree in Computer Science and the degree in Mathematics consist of 4 academic years with 10 courses each. The degree in Law consists of 4 academic years, the first one has 10 courses and the remaining 3 years have 8 courses each.

The values of the grades in the data set fall in the range between 0 and 10, although we have also missing data, indicated with *NaNs* (Not a Number). Our interpretation for a missing value (NaN) is that a student has not studied that particular courses yet. Whereas our interpretation for zeros is that a student has enrolled to that particular courses but has not completed the necessary tasks or exams to acquire a final grade.

2.1.1 Data Cleaning

To properly clean the dataset we have to take into account the amount of information that we can obtain from each variable. Table 2.2 shows the percentage of missing values of each variable:

Variables	Law	Computer Science	Mathematics
universitat procedencia	72.24%	34.6%	4.8%
sistema educatiu estranger	10.46%	35.3%	5.8%
priv pub secundaria	29.82%	97%	83%
tipus lloc secundaria	7.72%	97%	83%
lloc secundaria	10.40%	97%	83%
pais sistema estranger	98.04%	97%	98%
tipus lloc cfgs	93.86%	98%	99%
priv pub cfgs	91.98%	98%	99%
lloc cfgs	92.13%	98%	99%

Table 2.2: **Percentage of missing values of the variables**

If a variable has more than 50% of its entries as missing values then we consider that not enough information can be extracted from those entries and therefore it is removed from the dataset.

After removing the irrelevant variables of the dataset we want to remove the information of students who represent outliars. In order to do the following three criteria have been applied:

- Students with 5 or more missing values in an academic year are removed from the original data set.
- Students with a mean grade inferior to 2 points out of 10 in an academic year are removed from the original data set.
- Students who do not follow the standard enrollment procedure are removed from the data set. For instance, a student who enrolls to more than 10 courses in an academic year falls in this category.

All the data that have been removed from the original data set corresponds to rare cases that would bias the results of the models.

2.2 Dropout Prediction

We try to answer the following question: is it possible to use Machine Learning techniques to predict if a student will enroll to University in the second year given information of the first academic year?

As stated in [13], 58% of the dropouts occur in the first year of university studies. Thus we consider suitable to constraint our research to study first-year dropouts. All models could be trained to predict dropout for other years and different training sets.

Table 2.3 shows the percentage of students who dropped out after their first and second academic year in the UB data set for Law, Computer Science and Mathematics. As expected, the percentage in the first year is the highest.

	First year	**Second year**	**Total**
Law	14.6%	8.7%	23.3%
Computer Science	29.8%	13.8%	43.6%
Mathematics	39.4%	19.6 %	59.0%

Table 2.3: **Percentage of students how dropout by academic year and degree**

The dropout problem is an imbalanced binary classification problem which can be tackled by the following two-step procedure:

Step 1: Feature vector definition and data pre-processing. Each student in the data set is described using an (n+m)-dimensional vector consisting of the grades of each course of a given academic year. For Computer Science and Mathematics $n = 10$ for all the academic years and for Law $n = 10$ for the first academic year and $n = 8$ for the rest. m corresponds to the features explained in section 2.2.2.

Due to the nature of the problem, the training data has been balanced using Synthetic Minority Over-sampling Technique (SMOTE) [6] (see section 2.2.1) in a subset of the experiments.

Step 2: Classification. We train 5 classifiers: LR [22], GB [24], SVM [7], RF [16] and AdaBoost [15] using the feature vector of the training set samples. We choose these 5 classifiers since they are state of the art techniques which use different approaches to solve a classification problem.

The pipeline used to implement dropout prediction is shown in figure 2.1. Each step has been explained in the following sections.

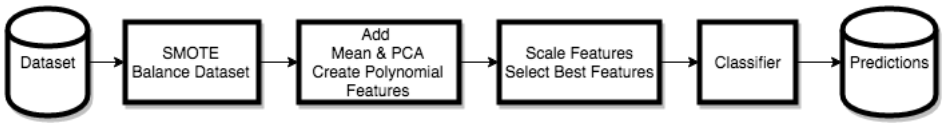


Figure 2.1: **Pipeline for dropout prediction**

2.2.1 SMOTE: Synthetic Minority Over-sampling Technique

Balancing a dataset before classification can yield to better results. This can be done in different ways:

1. Over-sampling with replacement: adding copies of existing samples to the dataset.
2. Over-sampling using SMOTE: creates new samples from the existing samples of the minority class
3. Under-sampling: randomly removing a subset of the majority class.

It is clear that when the percentage of samples of the minority class is specially low (10% or lower) such us the case of dropout, under-sampling alone is not an option.

We have chosen to balance our dataset by using over-sampling applying SMOTE and under-sampling.

In the next paragraph we provide a brief explanation of how SMOTE creates synthetic samples.

To create new representatives of the minority class (synthetic samples) SMOTE does the following: take a sample of the minority class (point O of figure 2.2) and compute its five nearest neighbors (points A to D). Draw the lines that join them with the sample point and then take a random point laying in each of the five lines (points A′ to D′). These new points will be added to the original dataset as new samples of the minority class.

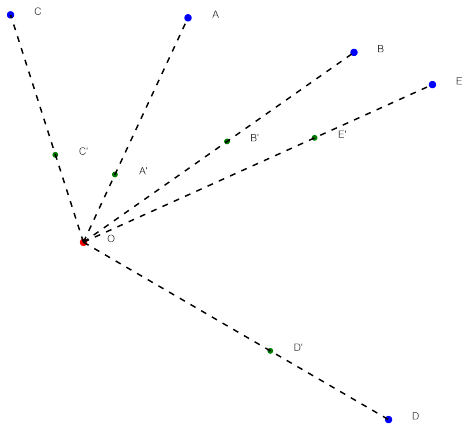


Figure 2.2: **Example of synthetic samples creation**

2.2.2 Feature Extraction and Selection

A key influential factor on the performance of a Machine Learning model is the feature vector space that this model has been trained on. In this section we explain what features we have found to be good candidates for predicting dropout intention.

We are looking for features that best split the samples of our datasets between dropout and non-dropout.

Working knowledge of the field suggests that the mean grade would be a good candidate. The figures 2.3, 2.4 and 2.5 show the mean grade of first-year students for each of the degrees depending on whether they dropout or not. The colours of the figures refere to the mean grade of the students (red mean grade inferior to 5, blue mean grade from 5 to 7, green from 7 to 9 and yellow from 9 to 10).

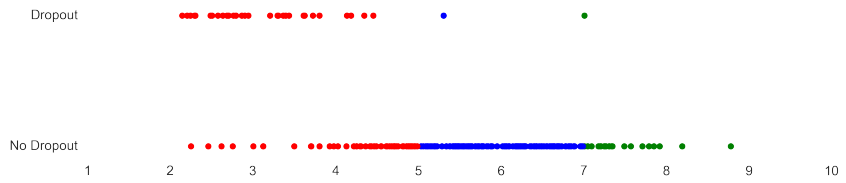


Figure 2.3: **Mean grade of first-year students for the degree in Computer Science**

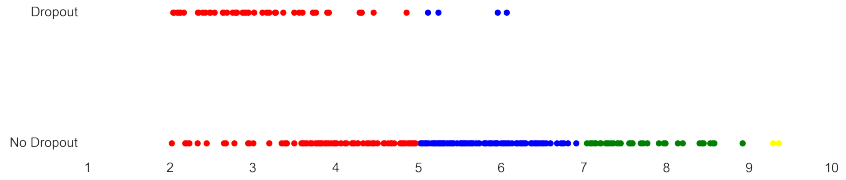


Figure 2.4: **Mean grade of first-year students for the degree in Mathematics**

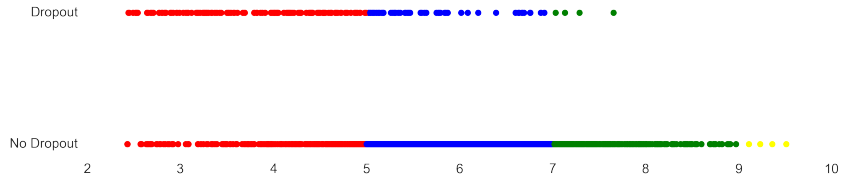
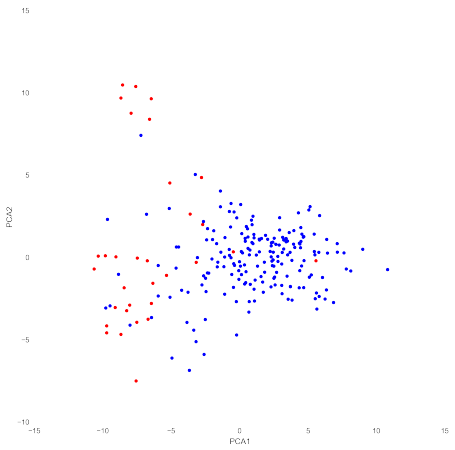


Figure 2.5: **Mean grade of first-year students for the degree in Law**

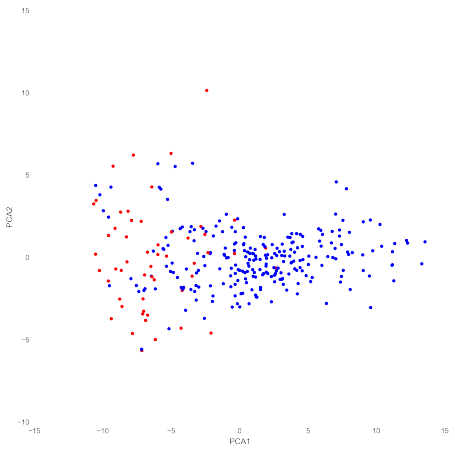
It can be seen that for the datasets corresponding to the degrees in Computer Science and Mathematics mean grade is a good feature to add to the feature vector space. However for the degree in Low this mean grade does not shed light to the problem of dropout prediction.

Another set of features that we want to explore are the Principal Component Analysis (PCA) components of the first-year grades. For visualization purposes we have limited

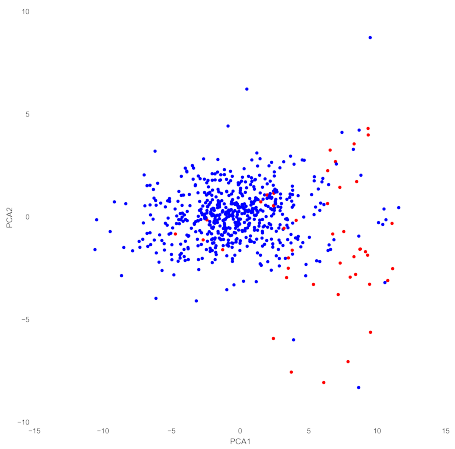
ourselves to 2 components. The figures 2.6 (a), 2.6 (b) and 2.6 (c) show the PCA decomposition for each of the degrees.



(a) Computer Science



(b) Mathematics



(c) Law

Figure 2.6: **PCA decomposition for the degree**

In this case we can see that the first component of the PCA decomposition could be added to the feature vector for all the datasets.

Sometimes combining features together can yield to new features with equal or better correlation with the prediction class. Taking this into consideration the feature vector space can be enlarged by computing polynomial combinations of the existing features.

We are going to limit our expansion of the dataset to a two-degree polynomial.

Let us define our feature space as F^n where n corresponds to the original features, that is, grades, mean grade and first PCA component. After computing the polynomial combination of these features, the new feature vector space space will be F^m where $m = \sum_{k=1}^{2} \binom{n+k-1}{k}$. We are going to denote this function by $poly : F^n \to F^m$

Using this procedure four different features vectors have been defined:

$$
\begin{aligned}
v_1 &= \{Grades\} \\
v_2 &= v_1 \cup \{mean\} \\
v_3 &= poly(v_2) \\
v4 &= poly(v_2 \cup \{PCA1\})
\end{aligned}
$$

2.2.3 Classification

In this section we provide a brief explanation of the five different models that have been used to perform dropout prediction.

Logistic Regression (LR)

LR is a linear model for classification. This model can be used in the case of binary or multiple classes. The explanation below has been restricted to the binary classification problem.

Let $x \in \mathbb{R}^n$ be a feature vector and let y be its class. LR transforms x to a real number by computing the following:

$$z = \alpha + \beta^T x$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^n$. These values are optimized by Maximum Likelihood or Stochastic Gradient Descent.

After projecting x into $\mathbb{R}$ the following probability is computed:

$$p = P(y = 1|z) = f(z)$$

where $f(z)$ is the logistic function (or sigmoid function) that in its standard form is:

$$f(x) = \frac{1}{1 + e^{-x}}$$

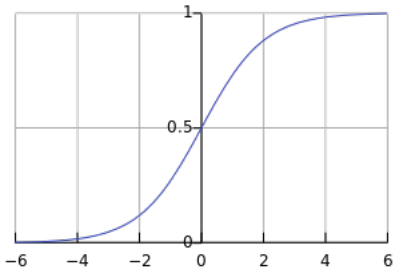


Figure 2.7: **Sigmoid function**

The prediction for feature vector x is given by:

$$\hat{y} = \begin{cases} 1 & p \geq 0.5 \\ 0 & p < 0.5 \end{cases}$$

Gaussian Naive Bayes Classifier (GB)

GB is a conditional probability model based in Bayes' theorem. Given a n-dimensional feature vector $(x_1, \ldots, x_n)$ and a classification class C, the algorithm computes $P(C|x_1, \ldots, x_n)$ using the Bayes' theorem. In practice, independence between features is assumed and the probability. Therefore given a classification class C_k, the probability of feature vector $(x_1, \ldots, x_n)$ belonging to this class is:

$$P(C_k|x_1, \ldots, x_n) \propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

The decision rule of the algorithm GB is:

$$\hat{y} = \underset{j \in \{1,\cdots,k\}}{\arg\max}\, p(C_j) \prod_{i=1}^{n} p(x_i|C_j).$$

For GB, the probabilities of each feature are computed as follows:

$$p(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{C_k}^2}} exp\Big(- \frac{(x_i - \mu_{C_k})^2}{2\sigma_{C_k}^2} \Big)$$

where, μ_{C_k} and $\sigma_{C_k}^2$ are the mean and the variance of the feature x_i associated with class C_k.

Adaptive Boosting (AdaBoost)

AdaBoost is a boosting technique which combines multiple weak classifiers h into a strong classifier H. A weak classifier is a model that performs slightly better than random guessing. The combination of T weak classifiers is performed as follows:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right),$$

where $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t} \right)$ and ϵ_t is the exponential loss function value for the weak learner h_t. We choose as weak classifier a *Decision Stump* [18], which is a one-level Decision Tree.

Support Vector Machines (SVM)

The explanation that follows has been restricted to two dimensional spaces and two classification classes for simplicity. All the definitions and examples have their analog in higher dimensional spaces.

To explain the model we will use the following definitions:

- Linearly separable set: A labeled set of points is said to be linearly separable if there exist at least one straight line that separate the points between classes.
- Maximum-margin line: Given a linearly separable set, define the set of all lines that separate the points into two clusters of classes as L. Then $l_M \in L$ is said to be the Maximum-margin line if it is at the same distance from the two clusters and this distance is maximal.

Given a linearly separable dataset, SVM solve the problem of finding the maximum-margin line. The figure 2.8 illustrates the problem:

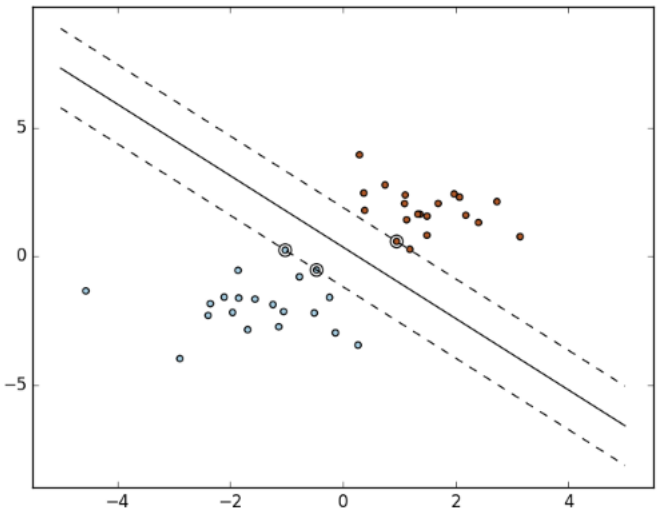


Figure 2.8: **Maximum-margin line example**

Random Forests Classifiers (RF)

Before explaining how RF works a brief explanation of Decision Trees is provided.

A Decision Tree is a set of decision rules over the features of a training set. The feature that best splits the samples of the training set is placed at the top of the tree. The decision rule associated with this feature is then used to separate the training set into two subsets and the same procedure is applied to those subsets. Each feature can be selected more than once.

To exemplify this, a simple Decision Tree has been grown from the data of the degree in Computer Science as shown in figure 2.9.

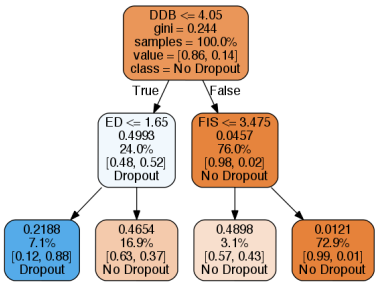


Figure 2.9: **Example of a simple Decision Tree**

A Random Forest classifier is an ensemble learning model that works by constructing a multitude of Decision Trees [23] and outputs the mode of the classes of the individual trees.

The performance of a RF classifier is better than that of a single Decision Tree because taking the mode of the outputs of the individual trees reduces the variance of the model. It is clear that if a major part of the Decision Trees were to be correlated this would not be true. To guarantee that the trees are the most decorrelated as possible two different techniques are used when training the RF:

1. Bootstrap aggregating (bagging): Suppose that we have a training set of n samples and we want to build a Forest with m trees. Each of these trees will be trained using a subset selected with replacement of the original training set.

2. Feature bagging: when growing a Decision Tree a random subset of the features is selected before each split.

2.2.4 Cross-valiation and Grid Search

All the models explained in the previous section have a finite set of parameters that can be adjusted to obtain their optimal performance for a given classification problem.

In Machine Learning, the process of finding those values is called parameter tuning and it is normally performed using what is called Grid Search.

Grid Search looks for the best combination of the parameters of a given model by training the model in each possible combination of those parameters and measuring its performance with a given metric. If this process is performed only on one single training set the results could be misleading. This is the reason for combining Grid Search with Cross Validation.

Cross Validation is used to evaluate the performance of a Machine Learning model on a given dataset. The dataset is first divided into n different subsets and then these subsets are rearranged to create n different folds. A fold consists in a training set and a test set. The training is formed by the union of $n-1$ subsets and the test set is the remaining set. This process is called n-Fold Cross Validation.

The creation of the folds is illustrated in figure 2.10.

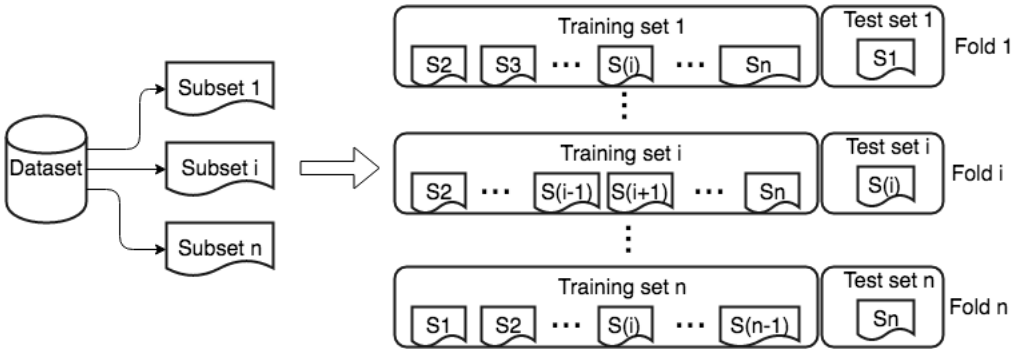


Figure 2.10: **Creation of n-folds**

Then the model is trained in each of the training sets and evaluated in the tests sets. The performance of the model is then the average of the performance in the tests sets.

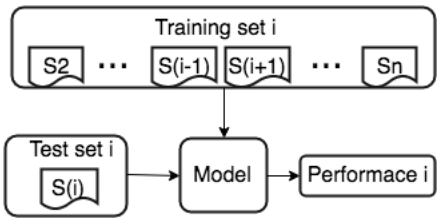


Figure 2.11: **Training and testing a model using fold** i

2.3 Final Grade Prediction

In this section we consider the problem of predicting final academic grades. In previous stages of the project, it was suggested that this problem could be tackled using a technique called Nearest Neighbours Collaborative Filtering in the following way:

Let us suppose that we have a database of n users and m items containing the rating that each user has given to a subset of the items. In general terms, Nearest Neighbours Collaborative Filtering predicts the ratings of the items that a user has yet to rate using information from other users.

If instead of a database containing ratings we have a database containing the final grades of n students for m different courses we can use the same technique to predict final grades of students.

The present work wants to expand the research done in this line by:

- Experimenting with more datasets.
- Implementing more complex versions of the algorithms used previously.

The rest of the section provides the theoretical framework on Recommender Systems used in this work.

2.3.1 Nearest Neighbours Collaborative Filtering

Nearest Neighbours Collaborative Filtering works under that similarity between users and items can be used to predict user ratings. There are two main blocks to make these predictions:

- **User-based**: In this approach, we suppose that similar users will rate items in similar ways. In the context of grades prediction, the key idea is that similar students should obtain similar grades.
- **Item-based**: Similar products should obtain similar opinions from users. In the context of grades prediction, the key idea is that students should perform in the same way for similar courses.

A more advanced approach is Collaborative Filtering with baseline adjustment. This approach combines the idea of Nearest Neighbours Collaborative Filtering with baseline predictors.

In the following subsections we explain how we have adapted Collaborative Filtering to the problem of grades prediction. We will explain common similarity measures and the approaches outline above.

In general, a recommendation system works by finding similarities between the rows of a sparse matrix and predicting the missing values using data from the same matrix. A Recommender systems works under the assumption that if two rows are similar and one of the rows is missing a value from a particular column it is valid to predict this unknown value using the value from the similar one.

2.3.2 Similarity measures

Suppose that we have two students which their grades are $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_n)$. The computation of their similarity can be done using Person Correlation:

$$s(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

There are a few considerations to have into account when using Pearson Correlation as a similarity measure.

Let us consider that the grades of a student has obtain the same grade for all the courses, i.e, $x = (x_1, \cdots, x_1)$. In this case the Person Correlation is not defined and another measure should be used.

An alternative choice is cosine similarity:

$$s(x,y) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

There is another important aspect to address to properly compute similarity between students (or courses).

Suppose that we want to compute the similarity between two student who have grades $x = (x_1, x_2, ?, \cdots, ?)$ and $y = (y_1, \cdots, y_n)$. In other words, the former student has only obtain two grades for courses 1 and 2 of a particular academic year and latter has obtained grades for all the courses. The similarity between this two students can only be computed using grades from courses 1 and 2. Therefore if $x_1 \approx y_1$ and $x_2 \approx y_2$ then $s(x,y) \approx 1$, which is not true at all. To solve this problem we have down-scaled the similarity between students by a suitable factor:

$$s_d(x,y) = s(x,y)\frac{min(c,n)}{n}$$

where c indicates the number of common courses that x and y have grades for. In the example above $c = 2$ and therefore $s(x,y) \approx \frac{2}{n}$. If c happens to be 0 it is obvious that the similarity between the two students can not be computed.

2.3.3 User and Item based Recommender Systems

In this section we are going to explain how User and Item based Recommender Systems can be used to predict grades.

For a given academic year of n courses, the grades of a particular student can be expressed using a n-dimensional vector such us:

$$s = (a_1, \cdots, a_j, \cdots, a_n)$$

Therefore the set of all the grades of all the students for a particular academic year can be denoted by a sparse matrix such as:

$$\begin{array}{c} \\ s_1 \\ \vdots \\ s_i \\ \vdots \\ s_m \end{array} \begin{array}{c} \begin{array}{ccccc} c_1 & \cdots & c_j & \cdots & c_n \end{array} \\ \begin{pmatrix} a_{11} & \cdots & a_{1j} & \cdots & ? \\ \vdots & & \vdots & & \vdots \\ ? & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \cdots & ? & \cdots & a_{mn} \end{pmatrix} \end{array}$$

User-based approach

With the notation above, let us work under the assumption that if two students have obtained similar grades in a given academic year, they should obtain similar grades in subsequent academic years. Therefore, we should be able to predict grades of students using grades from similar students.

Let us suppose that we have a matrix A containing the grades of m students for an academic year y of n courses. Let B be the matrix containing the grades of the m students for the academic year $y+1$. We can predict the grades for year $y+1$ of a studen given his/her grades of year y using matrices A and B by doing the following computation:

$$b_{ij} = \mu_i + \frac{\sum_{t \in S} s_{tj}^{\alpha}(b_{tj} - \mu_t)/\sigma_t}{\sum_{t \in S} |s_{tj}^{\alpha}|}\sigma_i \tag{2.1}$$

where S is a set consisting of the indices of the most similar students with student i. μ_i and σ_i are the mean and the standard deviation of student i.

Item-based approach

In this case the underling assumption is that if two courses are similar it should be possible to predict a missing grade from one of the courses using the grade of the other.

With the same notation above, predicting a grade for a course c_j of matrix B can be done by doing the same computation as in the User-based approach but working on the transposed matrices of A and B.

2.3.4 Collaborative Filtering with baseline adjustment

Collaborative Filtering with baseline adjustment works as follows:

A matrix R of size $c \times s$ is considered. c corresponds to the number of courses and s to the number of students. Then, the next baselines are computed:

$$b_{ui} = \mu + b_u + b_i$$

$$b_i = \frac{\sum_{u \in R_i}(r_{ui} - \mu)}{|R_i|}$$

$$b_u = \frac{\sum_{i \in R_u}(r_{ui} - \mu - b_i)}{|R_u|}$$

where, the set R_i consists of the students who have studied course i. Similarly, the set R_u consists of the set of courses studied by the student u. μ is the mean of the matrix R and r_{ui} is the value in position (u, i) of the matrix R.

To make a prediction for a student u and course i we use Equation (2.2):

$$r_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i,u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in S^k(i,u)} |s_{ij}|} \tag{2.2}$$

where the set $S^k(i, u)$ consists of the k courses studied by the student u that are most similar to the course i. This set of similar courses is computed by the KNN algorithm with Pearson similarity as distance measure. The similarity between a course i and a course j is denoted by s_{ij}.

2.4 Course Ranking

Using Collaborative Filtering we have been able to predict final academic grades. This is very rich information for tutors, however we find that a ranking of courses for each student can be even more readable and useful.

We want to give the tutor a criteria to identify what courses will be more difficult for a particular student and to do so, we do not need to know the exact grade for each course.

Given a grade g we apply standard Spanish thresholds to define four different discrete grades $A > B > C > D$:

$$f(g) := \begin{cases} D, & \text{if } g < 5 \\ C, & \text{if } 5 \leq g < 7 \\ B, & \text{if } 7 \leq g < 9 \\ A, & \text{if } 9 \leq g \leq 10 \end{cases} \tag{2.3}$$

We use these quantized grades to perform the ranking. Finally, we sort all courses of a student in descending order. With the new arrangement of the predicted grades, the tutor acquires extra information about a student at a glance. This can help the tutor to guide students using personalized information about them.

Chapter 3

Experiments

3.1 Evaluation Metrics

The performance of the classifiers is assessed using the standard measures of accuracy, recall, precision and F1. Mean Absolute Error (MAE) score and *Kendall* [19] ranking correlation are used to evaluate the recommender system.

3.1.1 Classifier metrics

The classifier metrics are defined as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn},$$

$$Precision = \frac{tp}{tp + fp},$$

$$Recall = \frac{tp}{tp + fn},$$

$$F1 = \frac{2tp}{2tp + fp + fn},$$

where tp is true positive (dropout), tn true negative (not dropout), fp false positive and fn false negative. We consider dropout as the positive class and non-dropout as the negative class. Because we want to minimize false negatives (students who drop out are predicted as students who do not drop out) we will select models with the high recall over those with better precision. We will analyze the trade-off between these metrics using F1.

3.1.2 Mean Absolute Error

To compute the difference between the matrix containing the real grades R and the matrix containing the predictions P, we use MAE:

$$MAE = \frac{1}{(c \times s) - R_{NaN}} \sum_{i=1}^{c} \sum_{\substack{j=1 \\ R_{i,j} \neq NaN}}^{s} |R_{i,j} - P_{i,j}|$$

where c is the number of courses, s the number of students and R_{NaN} is the number of missing values in matrix R.

3.1.3 Kendall Ranking Correlation

To evaluate the correlation between to rankings r_1 and r_2 we use *Kendall* τ *b* measure [19]:

$$\tau_{t,s} = \frac{P - Q}{\sqrt{(P + Q + T)(P + Q + S)}},$$

where P is the number of concordant pairs, Q is the number of discordant pairs, T is the number of ties only in r_1 and S is the number of ties only in r_2. If a tie occurs for the same pair in both r_1 and r_2, it is not added to either T or S.

A value close of 1 indicates strong correlation and -1 indicates strong disagreement between the rankings r_1 and r_2.

3.2 Dropout prediction

In this section we want to answer the following questions:

- **What is the best model to predict dropout intention for a given dataset?**

  To do so we will compare the performance of each model explained in 2.2.3 for the datasets of the degree in Mathematics, the degree in Computer Science and the degree in Law.

  We want good balance between precision and recall, therefore high F1-score would be preferable but we also want this balance to be tilted towards higher recall, i.e, we want the algorithm to be a bit pessimistic and predict dropout more often than not. This is the reason why we have divided the analysis of the best models into two parts, identifying the best model towards higher recall and identifying the best model towards higher F1.

- **What features should be used?**

  The models will be trained with each of the four feature vectors defined in section 2.2.2. We call the experiments using these features vectors **E1, E2, E3** and **E4** respectively. This will allow us to test if adding extra features to the feature vector space makes sense when it comes to predict dropout intention.

- **Does balancing the dataset previous classification improve recall?**

  As it has been mentioned before, misclassifying a student who has the potential of dropping out as one who does not is an error that should be avoided as much as possible. We think that balancing the dataset previous classification should improve recall of the models. To test this hypothesis, we will compare the performance of each of the model trained with balanced and unbalanced data.

Putting everything together, we will perform four experiments (E1 to E4) for each of the datasets explained in 2.1. In each of the experiments all the models explained in section 2.2.3 have been train with one of the feature vectors explained in 2.2.2 (E1 with v_1, etc). In addition to this, each experiment has been performed using balanced data with synthetic samples crated by SMOTE and unbalanced data. We well call *E1 S* experiment one with the balanced dataset and *E1 NS* the same experiment but performed under unbalanced data.

The training and tuning of the models have been done using Stratified 5-fold cross-validation and grid-search. More precisely, each dataset has been split five times into two subsets called train and test, the former containing 60% of the data and the latter 40%. Then the models have been trained and tuned using grid-search over the training set. Finally, the models have been validated using the left-out data in the test set. The values shown in the tables of the following subsections correspond to the metrics computed over the predictions made by the models on the test set.

The results of the experiments for each degree are shown in the following subsections.

3.2.1 Degree in Law

The table 3.1 shows the results of the experiments for the degree in Law. Each row of the table corresponds to one experiment. The experiment E_i S consists of evaluating all the models described in section 2.2.3 using feature vector v_i and balancing the dataset previous classification. The same experiment without balancing the dataset previous classification is denoted as E_i NS. The columns of the table show Recall and F1 score respectively for each model. The algorithm with best recall is highlighted in orange, the model with best F1 score is highlighted in blue and if in one experiment one model has both best recall and F1 score it is highlighted in green. The same notation and colour code have been used for the degree in Mathematics and Computer Science.

Experiment	LR		RF		SVM		GB		AdaBoost	
	RC	F1	RC	F1	RC	F1	RC	F1	RC	F1
E1 S	76	49	**83**	**76**	50	46	**86**	**66**	85	73
E1 NS	43	52	60	66	57	61	76	67	71	74
E2 S	76	49	62	59	49	44	79	47	67	54
E2 NS	39	50	50	60	56	61	67	52	57	66
E3 S	76	43	58	50	67	45	79	37	69	53
E3 NS	40	52	50	60	54	58	74	50	47	59
E4 S	21	5	54	49	78	39	69	43	58	40
E4 NS	44	53	46	56	54	61	67	52	49	59

Table 3.1: **Recall and F1 scores for experiments E1 to E4 for the degree in Law**

We can see that the best recall is 86% obtained by GB and the best F1 is 76% obtained by RF. The difference between the Recall of these two models is only 3% while the difference between F1 is 10%. The small gain in recall that GB provides is not enough to discard the 10% gain in F1 that RF reaches, therefore we consider RF to be the best model when predicting dropout for the degree in Law.

The table 3.2 shows the parameters selected after performing grid-search and the features that each model has selected when predicting dropout.

	Best Recall and F1
Model	RF
Parameters	criterion=entropy, maxdepth=none, k=10
Feature Vector	1
SMOTE	S
Score	83 76

Table 3.2: **Parameters selected after performing grid-search and the features that each model has selected when predicting dropout**

To finish our discussion on dropout prediction for the degree in Law we have plotted

the error distribution according to mean grade in figure 3.1.

In order to do so we have plotted the histogram shown in the next figure. The figure is divided in two parts. The left plot consists of the predictions made for students who do not drop out after studying their first academic year and the right plot consists of the predictions made for students who drop out after studying their first academic year. The light blue and red colors corresponds to the real distribution of students, whereas the dark colors corresponds to the distribution of students based on the predictions of the classifier. This visualization allows to clearly appreciate the FP errors as the light blue portions of the bars and the FN errors as the light red portions of the bars.

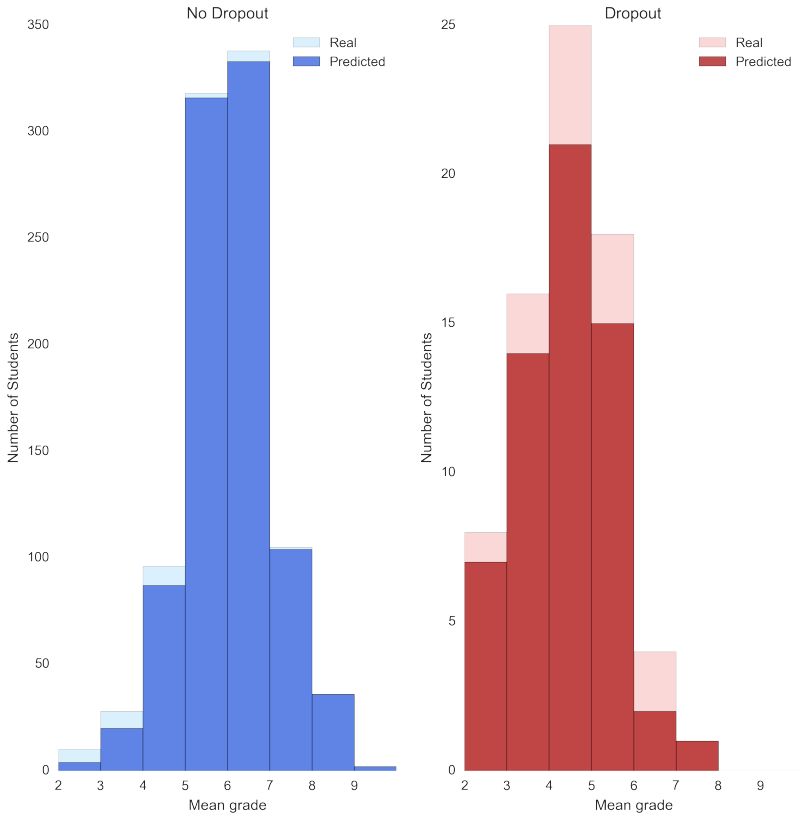


Figure 3.1: **Predictions of Random Forest for the degree in Law** Histogram showing dropout prediction for both students who do not drop out (blue) and students who drop out (red) grouped by their mean grade after their first academic year.

3.2.2 Degree in Computer Science

The table 3.3 shows the results for each experiment of the degree in Computer Science.

Exp	LR		RF		SVM		GB		AdaBoost	
	RC	F1	RC	F1	RC	F1	RC	F1	RC	F1
E1 S	83	59	67	57	83	59	83	62	67	57
E1 NS	67	64	50	57	**83**	**67**	83	65	42	45
E2 S	83	59	75	64	83	59	83	59	83	62
E2 NS	67	64	58	61	75	69	83	62	54	56
E3 S	83	61	83	65	83	63	83	59	83	62
E3 NS	**75**	**72**	67	64	83	62	83	61	67	57
E4 S	75	67	67	67	75	64	83	59	67	52
E4 NS	**75**	**72**	75	64	75	67	83	62	67	62

Table 3.3: **Recall and F1 scores for experiments E1 to E4 for the degree in Computer Science**

We can see that the best possible recall for the degree in Computer Science is 83%. This value is obtained in most of the experiments by different models but the model giving higher F1 score with this value of recall is SVM with a value of 67%.

The best possible F1 score is 72% obtained by LR using feature vector v_3 and v_4 both without balancing the dataset. This means that adding the first and second PCA components does not improve the results when it comes to F1 score for LR.

After this analysis we can conclude that the best models for predicting dropout for the degree in Computer Science are LR and SVM.

The table 3.4 shows the parameters selected after performing grid-search and the features that each model has selected when predicting dropout.

	Best Recall	Best F1
Model	SVM	LR
Parameters	k=9, C=10, $w = \{1:5\}$, kernel=rbf, $\gamma = 0.001$	k=6
Feature Vector	1	3
SMOTE	N	N
Score	83	72

Table 3.4: **Parameters selected after performing grid-search and the features that each model has selected when predicting dropout**

The distribution of the errors made by the selected models are plotted in figures 3.2 and 3.3.

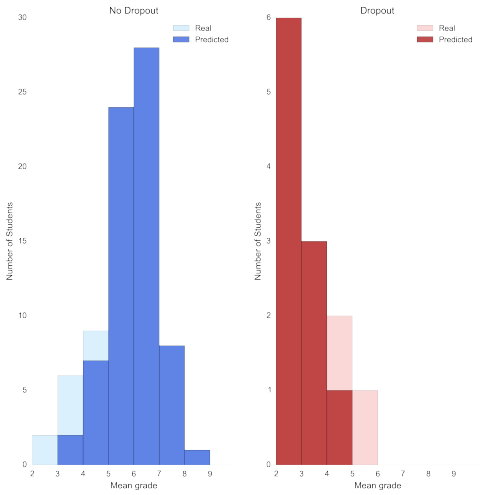


Figure 3.2: **Predictions of SVM for the degree in Computer Science** Plot showing dropout prediction for both students who do not drop out (blue) and students who drop out (red) grouped by their mean grade after their first academic year.

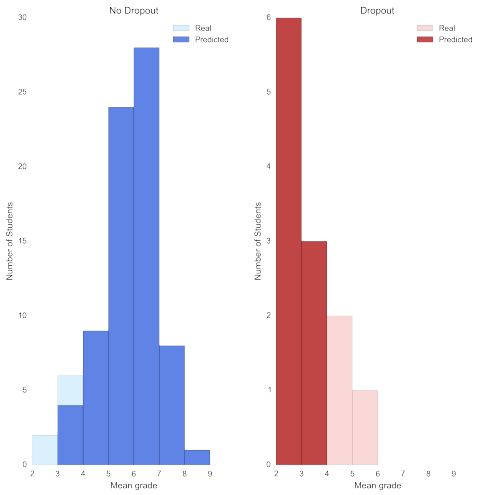


Figure 3.3: **Predictions of LR for the degree in Computer Science** Plot showing dropout prediction for both students who do not drop out (blue) and students who drop out (red) grouped by their mean grade after their first academic year.

3.2.3 Degree in Mathematics

The table 3.5 shows the results for each experiment of the degree in Mathematics.

Experiment	LR		RF		SVM		GB		AdaBoost	
	RC	F1	RC	F1	RC	F1	RC	F1	RC	F1
E1 S	74	49	53	45	68	58	79	53	58	49
E1 NS	63	57	63	59	68	54	74	55	58	55
E2 S	74	49	63	51	68	55	79	52	68	51
E2 NS	68	59	**68**	**60**	68	55	74	53	58	59
E3 S	74	52	68	52	79	55	84	53	63	51
E3 NS	68	58	53	56	68	52	84	53	58	58
E4 S	84	55	89	53	89	52	**89**	**56**	74	55
E4 NS	68	58	53	53	68	45	**89**	**56**	47	50

Table 3.5: **Recall and F1 scores for experiments E1 to E4 for the degree in Mathematics**

We can see that GB is the best algorithm to predict dropout in terms of recall in all the experiments. The performance of the algorithm increases significantly when polynomial features are introduced and reaches its highest value when the first and second PCA components are added to the feature vector space. A value of 89% is reached. We can see that after introducing polynomial features balancing the dataset does not have any effect on the performance of this algorithm.

The best algorithm regarding F1 is Random Forest obtaining a value of 60%.

The difference in F1 score of both algorithm is only 4% but the difference in recall is 21%. We think that given this situation the best algorithm to predict dropout for the degree in Mathematics is GB. As mentioned above, balancing the dataset does not change the performance of the model but it does increase training time, therefore we have chosen the best model to be GB trained with the feature vector v_4 with the original dataset.

The table 3.6 shows the parameters selected after performing grid-search.

	Best Recall & F1
Model	GB
Parameters	k=5
Feature Vector	4
SMOTE	N
Score	89 56

Table 3.6: **Parameters selected after performing grid-search and the features that each model has selected when predicting dropout**

The distribution of the errors made by the selected models are plotted in figures 3.2 and 3.3.

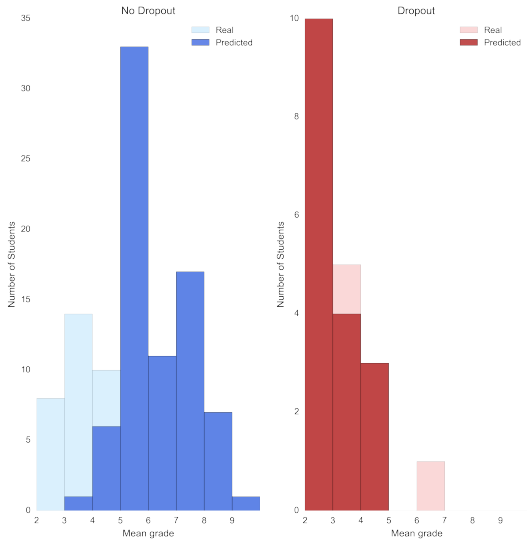


Figure 3.4: **Plot showing dropout prediction for both students who do not drop out (blue) and students who drop out (red) grouped by their mean grade after their first academic year.**

We can see that the distribution of the dataset allows the model to easily classify students with mean grade superior to 5. We consider that the most difficult classification of dropout intention is that of students with mean grade between 4 and 5. In this case, the model has been able to classify all the student that would dropout and 50% of the students who would not dropout. Regarding students with mean grades lower than 3, the model is really pessimistic and classifies almost all the students as dropout.

Finally, a permutation test over the tested models has been performed and the obtained p-value is 0.0099 for the three data sets, proving that the results are statistically significant.

3.2.4 Results Conclusion

- **What is the best model to predict dropout intention for a given dataset?**

  For the degree in Mathematics the best model is GB, with Recall of 89% and F1 of 56%.

  For the degree in Law the best model is RF, with Recall of 83% and F1 of 76%.

  For the degree in Computer Science it is not clear how to choose between SVM with 83% of Recall and 67% of F1 and LR with 75% of Recall and 72% of F1.

- **What features should be used?**

  For the degree in Mathematics the best performance has been obtained using feature vector v_4. The features selected as the best are the grade obtained in Linear Algebra, the mean grade, the first component of the PCA decomposition, the combination of ELPR and Arithmetic and the combination of LIRM and PCA1.

  For the degree in Law the best performance has been obtained using feature vector v_1 and all the features of the vector.

  For the degree in computer science the best recall has been obtained using feature vector v_1 and all the features while the best F1 score has been obtained using feature vector v_3 and the main features selected have been DDB and mean grade.

- **Does balancing the dataset previous classification improve recall?**

  Looking at table (3.7, Law) we can see that recall has been improved after balancing the dataset. However, for the other two degrees, recall has stayed the same in some of the experiments. The table 3.7 will allow us to understand why this is the case.

Dataset	Train		Test	
	Samples	Percentage	Samples	Percentage
Computer Science	20	14.8	12	13.3
Mathematics	34	18.9	19	15.8
Law	121	8	72	7

Table 3.7: **Percentage of dropout samples for each degree**

We can see that the Percentage of samples for the Degree in Law is significantly lower than that of the other degrees. The synthetic samples created using SMOTE allow the models to generalize better. For the other two degrees, the vast majority of the models have benefited from balancing the dataset but some of the models have not needed extra samples to be able to predict positive dropout correctly.

To sum up, from the results of the experiments in the degree in Mathematics and the degree in Law we could conclude that GB is the best model to predict dropout for smaller datasets and RF is better for much bigger datasets. However the results in the degree in Computer Science suggest that other algorithms have also the potential to be useful for dropout prediction. Under the context of this work, AdaBoost has

not performed well in the vast majority of the experiments and therefore should be descanted as a model to use in dropout prediction.

3.3 Grade prediction

In this section we want to decide which Recommender System from those explained in section 2.3 is the best to predict grades. We have compared the results of the Recommenders with more conventional methods such SVR and Linear Regression. The tables 3.8, 3.9 and 3.10 show the results obtained for each of the datsets.

Predictor	MAE	STD	CV Time
BaseLine	**1.349**	**1.187**	6.62s
User-Based	1.488	1.431	105.22s
Item-Based	1.422	1.275	15.31s
SVR	1.551	1.415	0.31s
LR	1.375	1.208	0.246

Table 3.8: **Prediction measures for the degree in Mathematics**

Predictor	MAE	STD	CV Time
BaseLine	**1.393**	**1.326**	8.56s
User-Based	1.427	1.364	146.82s
Item-Based	1.438	1.421	19.25s
SVR	1.446	1.468	0.35s
LR	1.401	1.352	0.18s

Table 3.9: **Prediction measures for the degree in Computer Science**

Predictor	MAE	STD	CV Time
BaseLine	**1.226**	**1.106**	42.81
Item-Based	1.330	1.226	143.54s
SVR	1.323	1.238	10.66s
LR	1.229	1.352	0.18s

Table 3.10: **Prediction measures for the degree in Law**

The Recommender System using baseline adjustment is the model giving the best performance for all the datasets. The analysis presented below corresponds to the predictions made by the Recommender.

It is interesting to notice that for large data sets we obtain a low MAE (1.226 for the degree of Law) and for smaller data sets a higher MAE is obtained (1.393 for Computer Science and 1.349 for Mathematics). In order to visualize how this error is distributed along the predictions, we visualize each predicted grade for second-year courses compared with the real grades of our data set in Fig 3.5 (Mathematics degree) and Fig 3.6 (Computer Science degree).

These figures are composed of two parts: 1) a central area showing a scatter plot of predicted grades (X-axis) and real grades (Y-axis) of all second year courses, along with a perfect score line (black line) and a best linear regression fitting line (blue line), and 2) two histogram plots showing the distributions of the predicted grades (X-axis) and real grades (Y-axis). The 4 shaded regions of the scatter plot correspond to the areas where the quantized predicted grades (Equation (2.3)), would be accepted as correct.

Let us comment the graphic in Fig 3.5. We observe that the linear regression fitting line is near to the perfect score line. The points falling in the white areas of the plot are those wrongly predicted quantized grades. To provide more visual information we have colored each point according to the mean grade obtained by the student in the previous academic year. Each color corresponds to A = yellow, B = green, C = blue, D = red, following thresholds created by Equation (2.3).

It can be seen that the vast majority of the grades that should fall within the red-shaded region of the plot do so. This means that our recommender system is able to identify the courses that will be the most difficult ones for a student in the next academic year. This is satisfactory, since it is particularly important to rank the most difficult courses properly, where tutor can influence. The dots drawn in the left of each shaded region correspond to courses with a predicted grade lower than the real one while the dots drown in the right hand side of each shaded region correspond to courses with a predicted grade higher than the real one. The number of dots in the first region compared to the number of dots in the second region and the proximity of those to the shaded regions indicates that the model is moderately pessimistic. This characteristic is essential due to the nature of the problem in hand. We prefer to give extra support to a student that would successfully pass a course without help than having the risk of missing a student in need of advice.

We finish the discussion of our predictions by analyzing the distribution of the data set. The histogram plotted along the Y-axis in Fig 3.5 shows that there are fewer examples of courses which have a grade inferior to 5 than the rest. This explains the tendency of our recommender to give predictions closer to 5.

Fig 3.6 show the performance of the recommender when predicting grades for smaller data sets. The analysis is done with the data of the degree in Computer Science. Note that the distribution of the data in this plot is more skewed than the previous one (Fig 3.5), making the predictions of grades even more challenging. It can be seen that the prediction in the red-shaded area is more poorly done than before. The reason for this is that data does not contain many samples of grades between 5 and 0 as shown in the Y-axis histogram. Thus, the recommender does not have enough information to perform a more accurate approximation and it tends to predict following a Gaussian distribution, as shown in the X-axis histogram. Moreover, looking at the colors of the points, one can appreciate that the wrongly predicted grades of the red-shaded area correspond to students who did relatively well in their previous academic year (blue points) and the correctly

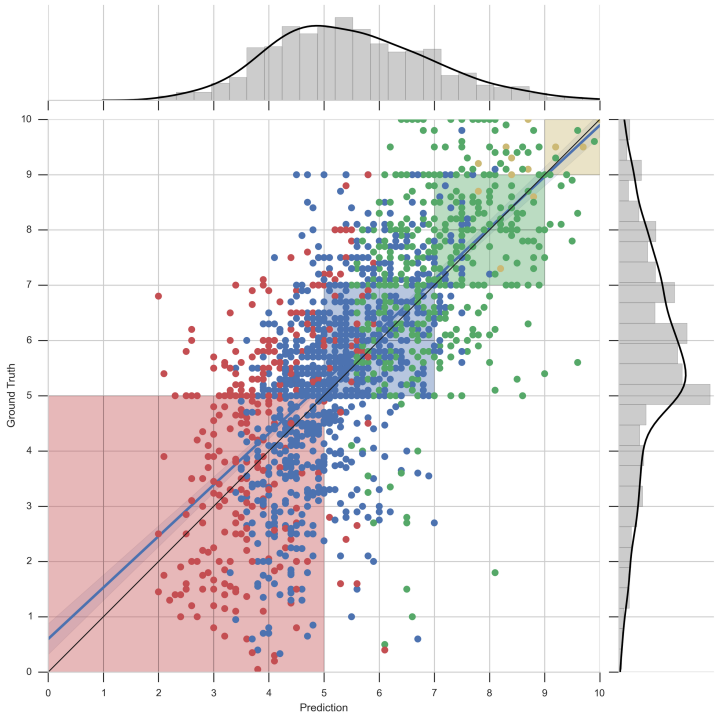


Figure 3.5: **Scatter plot and distribution visualizations of grade predictions - Mathematics.** Predicted values against real values for second-year grades for the degree in Mathematics. Each point corresponds to a grade of a student for a particular course. The dots are colored accordingly to the mean grade obtained by the students in the previous academic year. The shaded regions correspond to acceptable errors. The histogram plots show the distributions of the predicted grades (X-axis) and real grades (Y-axis).

predicted ones correspond to those students who did badly in their previous year (red points).

We want to provide the tutor with a way to further interpret the data provided by our recommender. Fig 3.7 shows the spread in the error between the predicted grades and the real ones for the degree in Mathematics. We compute the difference between predicted grades and real grades. A positive value of the difference means that the recommender has predicted a higher grade than the real one and a negative value means the opposite. This figure shows how to interpret a prediction of a given value made by the recommender. For instance, if the recommender has predicted a 10, this value is likely to be any number between 8 and 10 since the deviation is 2. We can observe that the most accurate predictions are those made for grades higher than 5, being 6 the most unreliable prediction of all (presence of outliers).

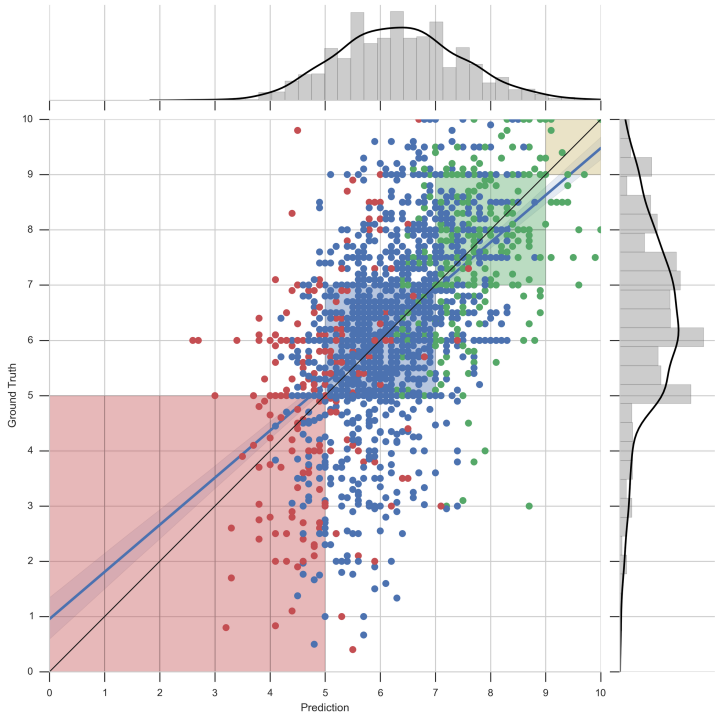


Figure 3.6: **Scatter plot and distribution visualizations of grade predictions - Computer Science.** Predicted values against real values for second-year grades for the degree in Computer Science. Each point corresponds to a grade of a student for a particular course. The dots are colored accordingly to the mean grade obtained by the students in the previous academic year. The shaded regions correspond to acceptable errors.

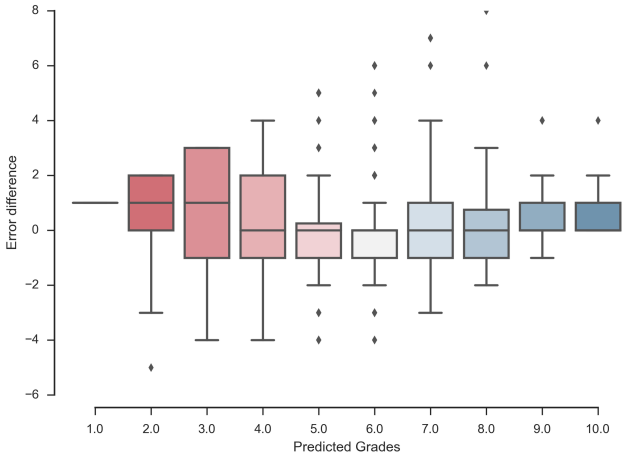


Figure 3.7: **Interpretation graphic for predicted grades errors - Mathematics.** Box plot showing the error difference made by the recommender regarding predicted grades of the degree in Mathematics.

Finally, we compute the Kendall measure and provide another way to elucidate the ranking performance.

A Kendall correlation score of 0.29 is obtained for the degree in Computer Science, meaning that there is moderate agreement between the predicted ranking and the real one. A Kendall correlation score of 0.12 and 0.21 for the degrees in Law and Mathematics respectively.

The heat map shown in Fig 3.8 provides a way to interpret the meaning of the obtained Kendall correlation score. The X-axis corresponds to the positions of the predicted ranked courses and the Y-axis corresponds to the real ranked course positions. The intensities of the heat map (in the scale shown in the colorbar of the plot) have the following meaning: The intensities in the diagonal cells illustrates the percentage of the correctly positioned courses. The rest of cells illustrates the error in the ranking. For instance, we can see that in position (1,1) the intensity of the cell corresponds to a value of almost 0.40, which means that in average, 40% of the courses that where predicted to be at the top of the ranking (position 1) were actually at the top in the real ranking (position 1). In other words, given a new ranking, there is a probability of 0.40 that course placed in the first position is predicted correctly. Moreover, the numeric values of the diagonal correspond to the average error for each position in the ranking, i.e, the value 1.8 in position (1,1) means that in average a course ranked in position 1 in the predicted ranking would be likely to be at position 3 in the real one. It is worth to note that the intensities of cells in position 1 and 2 are much higher than the rest. Taking into account that the mean average deviation for positions (1,1) and (2,2) is almost 2 we can deduce that it is likely that the two most difficult courses for a student will be among the top 4 of the predicted ranking. Therefore, the tutor can fairly advise the students to focus on the top 4 ranked courses.

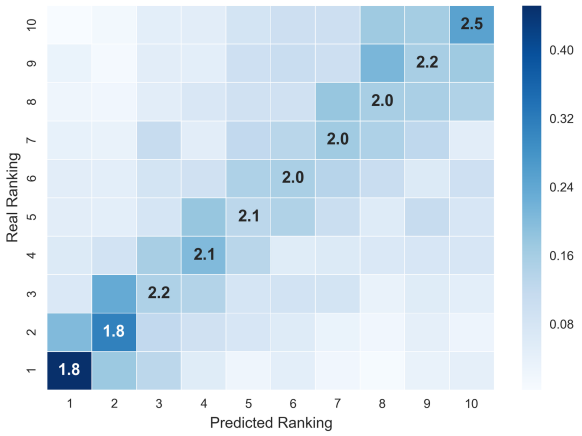


Figure 3.8: **Heat map for ranking evaluation - Computer Science.** Heat map showing probabilities of ranking correctness for the degree in Computer Science.

3.4 Results Discussion

We have seen that using Collaborative Filtering with baseline adjustment has proven to provide the best results in final grade prediction. The MAE scores have been 1.349, 1.393 and 1.226 for the degrees in Mathematics, Computer Science and Law respectively. Using the presented visualization tools, the tutors could obtain relevant information about the students and provide them with more informed guidance. However, we consider that this results are not sufficient to build a tool such the one outlined in section 4 and further research should be done to improve the performance of the presented models.

Chapter 4

Tool Design

In this final section we want to outline a possible design of the tutoring tool. We consider that the this tool should accomplish two main tasks:

- Provide the tutor with periodic notifications of students' performance. This could be performed by predicting dropout intention every time new academic data from the students is obtained.
- Provide the necessary resources to the students in most need of help. Predicting grades of the students and sending resources for those subjects that the system has predicted as most difficult for them could accomplish this task.

4.1 Internal Structure

The tutoring tool could be build on top of the following three modules:

1. **Training Data**

   This first module collects all the data that has been selected for training the models used by the other two modules. As mentioned in the introduction, different types of data could be added to the models in order to perform better predictions.

   - Grades of the students
   - Students' motivation
   - Opinion polls

2. **Dropout prediction**

   A good indicator of student's performance is dropout intention. This module provides a prediction of dropout intention based on the student's grades. This model would be build around a particular Machine Learning model.

3. **Grades prediction**

   This module will provide the prediction of grades for each student.

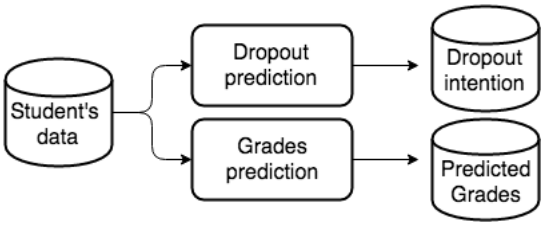


Figure 4.1: **Activation of the system's prediction**

4.2 System activation

In order to gain the most from the system, the tutor and the students should be able to obtain accurate information from the it as many times as possible.

The next figure shows a time line of the first three years of a standard four-year degree. Each blue vertical line marks the moment when new information about the students could be obtain and therefore new predictions could be made. At each of those moments the system will predict dropout intention and the grades for the courses that the students will study until the next prediction can be made.

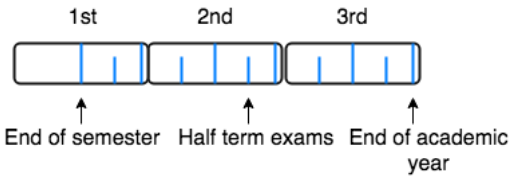


Figure 4.2: **Activation of the system's prediction**

In the first academic year the system does not have information about the student, therefore no prediction can be made. At the end of the first semester of the first predictions can be made. Then the next prediction would be after half term exams and the last one at the end of the academic year. For the second year, the first prediction would be given after the first half term exams and so on.

After each prediction is made, the information is given to the tutor using the interface described in the next section. Each student that the system has predicted that is in danger of failing a particular subject would receive additional information and resources for that particular subject.

4.3 Tutor interface

The tutor interface has been designed to provide as much information as possible in order to help the tutor better track students' performance and progression throughout their university studies.

The interface is build around blocks like the one shown in the figure 4.3.

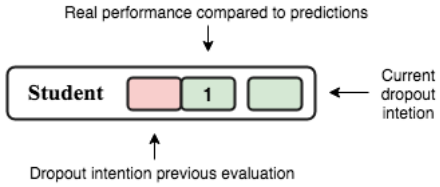


Figure 4.3: **Information of the student in the tutoring interface**

This blocks consists of the following three parts:

1. Unique identifier of the student (NIUB, DNI, Name)
2. Previous evaluation: this second part consists of the dropout intention of the student of the previous evaluation and the real performance of the student compared to the predictions.
3. Current evaluation: the last component of the block consists of the current dropout intention of the student.

We now provide a commented example of the interface.

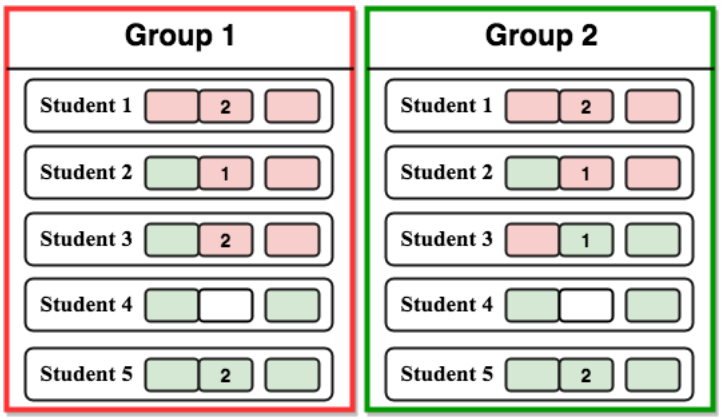


Figure 4.4: **Tutor interface**

In the example shown in the figure 4.4 the tutor is in charge of two groups of five students each.

At first glance the tutor can see that the frame of Group 1 is red, meaning that more than 50% of the students in danger of dropping. Let us analyze the students of the first

group. They have been sorted by dropout intention and identifier. The tutor can see that the first student was in danger of dropping out and performed 2 points below the predicted mean grade average in the previous evaluation. The system now predicts that this student is again in danger of dropping out. The second student was not predicted to be in danger but the student performed worse than expected and the system now predicts that he/she is in danger of dropping out. The same happens with Student 3. Student 4 was not in danger and performed accordingly the predictions, shown by a white rectangle. The system predicts that this student continues to not being in danger of dropping out. Student 5 is similar to Student 4 but he/she has performed better than the predictions indicated.

In more general terms, the tutor can see that in the previous evaluation only one student was in danger of dropping out but now two 3 students are in danger.

The analysis of Group 2 is the same but in this case less than 50% of the students are in danger of dropping out, therefore the frame of the group is green.

After this analysis the tutor could conclude that three students from group 1 would benefit from extra help. This help could be provided by a personal interview with the tutor alongside different resources in form of books, online courses and others.

The tutor could benefit from more concrete information of each student when making the decision of what resources and advice provide to the students.

The figure 4.5 shows a possible interface for each student.

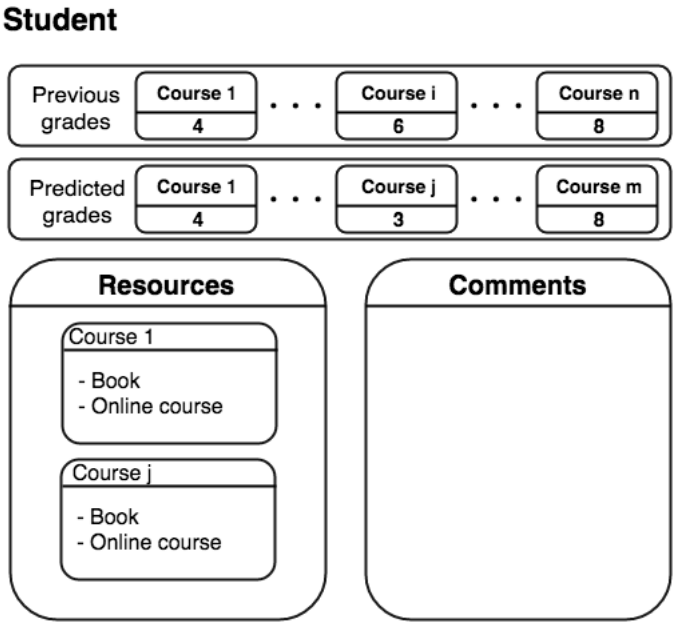


Figure 4.5: **Personal information of each student provided by the system**

In the previous figure we can see three different parts. The first part gives information about the grades of the student. The grades of the previous evaluation are shown first and the predicted grades for the current evaluation are shown second. The part titled

resources contains the most relevant resources that the tutor can provide to the student for the subjects that the system predicts as failed. A section where the tutor can add different comments about the students is also interesting to add to the system.

Chapter 5

Conslusions

In this work, we have presented a data-driven system to help tutors in the early detection of dropout and the prediction of courses grade as well as courses ranking. We have compared different machine learning methods and selected the ones with the best performance. The results in dropout prediction are promising, obtaining a Recall score of 83%, 83% and 89% for the degrees in Computer Science, Law and Mathematics respectively. We conclude that Gaussian GB could be a good model to predict dropout intention for small datasets and RF for bigger datasets. However, the results in the degree in Computer Science have shown that other models such us SVM and LR should not be descanted us candidates for dropout prediction until further research is done on this topic.

After adding the mean grade of the students and the component of greater variance of the PCA decomposition of the grades, we have seen improvements to dropout intention prediction for the degree in Computer Science and the degree in Mathematics but not for the degree in Law.

Our experiments have shown that balancing the datasets previous classification using SMOTE have improved the performance of the models in the vast majority of cases.

Regarding grade prediction, our recommender system is able to predict final grades with a mean absolute error of 1.22, 1.39 and 1.35 for the degrees in Law, Computer Science and Mathematics respectively.

In order to complete the evaluation of the performances we have developed visualization tools to better understand the obtained results. In particular, these visualizations allow to interpret: where the system commits errors on dropout prediction for each degree; how the errors of predicted grades are distributed for each degree; and how correct the ranking is for each degree.

This work proves the power of machine learning techniques in dropout prediction. This complements previous works done in the Educational Sciences community, where Statistical approaches are mainly used for understanding the underlying cause of problems such as dropout intention.

Regarding educational implications, our system can be extremely useful for the tutors, which will be able to know beforehand which students need help and in which subjects. This information will assist tutors in their main task, which is the personalized enrollment guidance and orientation. Moreover, the tutor task will be more guided by means of the

presented visualization tools. We expect that this aided tutorial system has an impact in the student motivation, satisfaction and results improvement. In this way, dropout intention could be reduced and student engagement could be increased.

As future work, improving grades prediction results should be a priority. This could be done by using hybrid and SVD Recommender Systems. Adding additional features to the datasets could help. Interesting options would be the number of times that a student has enrolled to a particular subject, information about the teacher giving that particular course and motivation of the student. We think that this additional features could also help to improve the results of dropout prediction.

In order to improve the quality of the conclusions drawn on this study the number of students, variety of degrees and universities should be increased.

Analyzing student's profiles by means of different clustering techniques would help to better identify general characteristics of the students. The results could help to better understand and improve the predictions of the models proposed in this work.

Finally, implementing a working prototype of the system and testing it with new data would help to add new features and useful information that the tutors could use to provide the students with better insight. Building a database of resources for each of the subjects of the studied degrees would also help to facilitate the task of the tutors.

Chapter 6

Acknowledgments

This work was supported by Spanish Ministry of Science and Innovation (Grant TIN2013-43478-P and Grant TIN2015-66951-C2-1-R), by Catalan Government award 2014 SGR-1219 and SGR-561 and by University of Barcelona (Grant 2014PID-UB/068 and Grant REDICE-1602 of the INDOMAIN Innovation and Teaching Group).

I would like to thank all the people who has collaborated to make the present research possible. First, Dra. Igual for her professional guidance. Without her advice this TFG would not have been possible. Carme Zambrana for her critic insight and motivation. I also want to thank Marc Beltrán for his invaluable opinion on visualizations. Finally, I would like to thank my mother and grandmother for they unconditional support and love.

Bibliography

[1] *Education at a glance: Oecd indicators*, http://www.oecd.org/education/eag2013.htm, 2013.

[2] *Indicators for educational development and analysis of qualifications*, http://winddat.aqu.cat, 2016.

[3] F Belloc, A Maruotti, and L Petrella, *How individual characteristics affect university students drop-out: a semiparametric mixed-effects model for an Italian case study*, Journal of Applied Statistics Month **00** (2010), no. 0, 1–15.

[4] Ana Belén Bernardo-Gutiérrez, Rebeca Cerezo-Menéndez, José Carlos Núñez-Pérez, Ellian Tuero-Herrero, and María Esteban-García, *Predicción del abandono universitario: variables explicativas y medidas de prevención*, Revista Fuentes **0** (2016), no. 16.

[5] Ana Casaravilla, *El abandono académico: análisis y propuestas paliativas. dos proyectos de la universidad politécnica de madrid*, Revista Pensamiento Matemático **4** (2014), no. 1, 7–15.

[6] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, *Smote: Synthetic minority over-sampling technique*, Journal of Artificial Intelligence Research **16** (2002), 321–357.

[7] Nello Cristianini and John Shawe-Taylor, *An introduction to support vector machines: And other kernel-based learning methods*, Cambridge University Press, New York, NY, USA, 2000.

[8] Hans de Wit, *European integration in higher education: The bologna process towards a european higher education area*, pp. 461–482, Springer Netherlands, Dordrecht, 2007.

[9] Lola C. Duque, *A framework for analysing higher education performance: students' satisfaction, perceived learning outcomes, and dropout intentions*, Total Quality Management & Business Excellence **25** (2014), no. 1-2, 1–21.

[10] Lola C. Duque, Juan C. Duque, and Jordi Suriñach, *Learning outcomes and dropout intentions: an analytical model for spanish universities*, Educational Studies **39** (2013), no. 3, 261–284.

[11] N. Dávila, Mª D. García-Artiles, J. Mª Pérez-Sánchez, and E. Gómez-Déniz, *An asymmetric logit model to explain the likelihood of success in academic results*, Revista de Investigación Educativa **33** (2015), 27–45.

[12] María Esteban-García, Ana Belén Bernardo-Gutiérrez, and Luis J. Rodríguez-Muñiz, *Permanencia en la universidad: la importancia de un buen comienzo*, Aula Abierta **44** (2016), no. 1, 1 – 6.

[13] Joaquín Gairín, Xavier M. Triado, Mònica Feixas, Pilar Figuera, Pilar Aparicio-Chueca, and Mercedes Torrado, *Student dropout rates in catalan universities: profile and motives for disengagement*, Quality in Higher Education **20** (2014), no. 2, 165–182.

[14] Hebe Goldenhersh, Adela Coria, and Martín Saino, *Deserción estudiantil: desafíos de la universidad pública en un horizonte de inclusión Deserción*, RAES Revista Argentina de Educación Superior **3** (2011), 96–120.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[16] Tin Kam Ho, *Random decision forests*, Proceedings of the Third International Conference on Document Analysis and Recognition (Washington, DC, USA), ICDAR '95, vol. 1, IEEE Computer Society, 1995, pp. 278–282.

[17] M. Huberth, P. Chen, J. Tritz, and Mckay TA., *Computer-tailored student support in introductory physics*, PLoS ONE **10** (2015), no. 9.

[18] W. Iba and P Langley, *Induction of one-level decision trees*, Proceedings of the Ninth International Conference on Machine Learning, 1992, pp. 223–240.

[19] William R. Knight, *A computer method for calculating Kendall's tau with ungrouped data*, Journal of the American Statistical Association **61** (1966), no. 314, 436–439.

[20] G. McKinney and A. Gunawardana, *Evaluating recommendation systems*, Recommender Systems Handbook (2011), 257–297.

[21] Claude Montmarquette, Sophie Mahseredjian, and Rachel Houle, *The determinants of university dropouts: a bivariate probability model with sample selection*, Economics of Education Review **20** (2001), 475–484.

[22] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll, *An introduction to logistic regression analysis and reporting*, The Journal of Educational Research **96** (2002), no. 1, 3–14.

[23] J. R Quinlan, *Induction of decision trees*, Machine Learning **1** (1986), 81–106.

[24] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, *Tackling the poor assumptions of naive bayes text classifiers*, Proceedings of the Twentieth International Conference on Machine Learning, 2003, pp. 616–623.

[25] David Rodríguez-Gómez, Mònica Feixas, Joaquín Gairín, and José Luís Muñoz, *Understanding catalan university dropout from a cross-national approach*, Studies in Higher Education **40** (2015), no. 4, 690–703.

[26] L. Igual S. Rovira and E. Puertas, *Data-driven system to predict academic grades and dropout*.

[27] Alex J. Smola and Bernhard Schölkopf, *A tutorial on support vector regression*, Statistics and Computing **14** (2004), no. 3, 199–222.

[28] A. Tekin, *Early prediction of students' grade point averages at graduation: A data mining approach*, Eurasian Journal of Educational Research **54** (2014), 207–226.

[29] Vincent Tinto, *Research and Practice of Student Retention: What Next?*, Journal of College Student Retention **8** (2007), no. 1, 1–19.

[30] C. Van Soom and V. Donche, *Profiling first-year students in stem programs based on autonomous motivation and academic self-concept and relationship with academic achievement.*, PLoS ONE **9** (2014), no. 11.

[31] Kelly H. Zou, Kemal Tuncali, and Stuart G. Silverman, *Correlation and simple linear regression.*, Radiology **227** (2003), no. 3, 617–622.